WHAT IS CLAIMED IS:

1. A method of organizing components to provide access to a service, comprising:

   grouping components together to perform the service, wherein each component implements an interface for communicating with an assembly manager; and

   defining an assembly, the assembly having a name and an assembly definition having metadata information identifying each component in the group of components and any further interfaces implemented by or used by any of the components, whereby the assembly definition is configured to be loaded into the assembly manager.

2. The method of claim 1, further comprising associating the name of the assembly with a role name associated with the service.

3. The method of claim 1, further comprising:

   grouping the assembly with components to perform a second service; and

   defining a second assembly, the second assembly having a second name and a second assembly definition having metadata information identifying the assembly and each component in the group of components and any further interfaces implemented by or used by the assembly and any of the components, whereby the second assembly definition is configured to be loaded into the assembly manager.

4. The method of claim 1, further comprising:

   modifying one of the components;

   creating a new component to filter information that passes through an interface connected to the modified component; and

   modifying the assembly definition to specify the new component and an interface connecting the new component to the modified component, whereby the modified component and new component so connected produce filtered information compatible with other components in the assembly.

5. The method of claim 4, wherein modifying the component alters the processing of information and renders the modified component and information incompatible with the other components in the assembly.

6. The method of claim 1, wherein defining the assembly comprises identifying client and server relationships between the components and interfaces.

7. The method of claim 1, wherein the assembly definition is represented using Extensible Markup Language (XML).

8. The method of claim 1, wherein the components and interfaces comply with an object model architecture.

9. The method of claim 8, wherein the object-model is selected from a set of object-models including Component Object Model (COM), Bravo Interface Binder (BIB), and Common Object Request Broker Architecture (CORBA).

10. A method of providing access to a service by a component-based application, comprising:

receiving a request from the component-based application that identifies a service;

accessing an assembly definition associated with the service and having metadata information specifying a number of components used to perform the service and interfaces implemented by and used by the components;

loading each component identified in the assembly data-structure into an area for processing; and

connecting an interface associated with each loaded component to other components according to the meta-data information in the assembly definition to form an assembly, whereby the application has access to an interface for communicating with the assembly.

11. The method of claim 10, further comprising:

connecting interfaces identified in the assembly definition to the loaded components; and

connecting interfaces associated with components in the assembly definition but not identified in the assembly definition to the loaded components.

12. The method of claim 11, further comprising connecting interfaces in the assembly to components in a previously loaded assembly.

13. The method of claim 10, further comprising:

receiving an indication that the access to the requested service is not longer required;

disconnecting the interface from each component associated with the requested service; and

unloading each disconnected component and the corresponding assembly definition while the component-based application remains loaded.

14. A method for gaining access to a service, comprising:

identifying a service for processing data;

calling an assembly manager with a service request corresponding to the service; and

accessing an assembly capable of performing the service, the assembly including components and interfaces specified in an assembly definition and loaded by the assembly manager.

15. The method of claim 14, wherein the service request comprises a name associated with the assembly definition.

16. The method of claim 14, wherein the service request comprises a role name associated with the service.

17. A computer program product, tangibly stored on a computer-readable medium, for organizing components to provide access to a service, the product comprising instructions operable to cause a programmable processor to:

group components together to perform the service, wherein each component implements an interface for communicating with an assembly manager; and

define an assembly, the assembly having a name and an assembly definition having metadata information identifying each component in the group of components and any further interfaces implemented by or used by any of the components, whereby the assembly definition is configured to be loaded into the assembly manager.

18. The product of claim 17, further comprising instructions operable to cause the processor to associate the name of the assembly with a role name associated with the service.

19. The product of claim 17, further comprising instructions operable to cause the processor to:

group the assembly with components to perform a second service; and

define a second assembly, the second assembly having a second name and a second assembly definition having metadata information identifying the assembly and each component in the group of components and any further interfaces implemented by or used by the assembly and any of the components, whereby the second assembly definition is configured to be loaded into the assembly manager.

20. The product of claim 17, further comprising instructions operable to cause the processor to:

modify one of the components;

create a new component to filter information that passes through an interface connected to the modified component; and

modify the assembly definition to specify the new component and an interface connecting the new component to the modified component, whereby the modified component and new component so connected produce filtered information compatible with other components in the assembly.

21. The product of claim 20, wherein modifying the component alters the processing of information and renders the modified component and information incompatible with the other components in the assembly.

22. The product of claim 17, wherein the instructions operable to cause the processor to define the assembly comprise instructions operable to cause the processor to identify client and server relationships between the components and interfaces.

23. The product of claim 17, wherein the assembly definition is represented using Extensible Markup Language (XML).

24. The product of claim 17, wherein the components and interfaces comply with an object model architecture.

25. The product of claim 24, wherein the object-model is selected from a set of object-models including Component Object Model (COM), Bravo Interface Binder (BIB), and Common Object Request Broker Architecture (CORBA).

26. A computer program product, tangibly stored on a computer-readable medium, for organizing components to provide access to a service by a component-based application, the product comprising instructions operable to cause a programmable processor to:

receive a request from the component-based application that identifies a service;

access an assembly definition associated with the service and having metadata information specifying a number of components used to perform the service and interfaces implemented by and used by the components;

load each component identified in the assembly data-structure into an area for processing; and

connect an interface associated with each loaded component to other components according to the meta-data information in the assembly definition to form an assembly, whereby the application has access to an interface for communicating with the assembly.

27. The product of claim 26, further comprising instructions operable to cause the processor to:

connect interfaces identified in the assembly definition to the loaded components; and

connect interfaces associated with components in the assembly definition but not identified in the assembly definition to the loaded components.

28. The product of claim 27, further comprising instructions operable to cause the processor to connect interfaces in the assembly to components in a previously loaded assembly.

29. The product of claim 26, further comprising instructions operable to cause the processor to:

receive an indication that the access to the requested service is not longer required;

disconnect the interface from each component associated with the requested service; and

unload each disconnected component and the corresponding assembly definition while the component-based application remains loaded.

30. A computer program product, tangibly stored on a computer-readable medium, for gaining access to a service, the product comprising instructions operable to cause a programmable processor to:

identifying a service for processing data;

calling an assembly manager with a service request corresponding to the service; and

accessing an assembly capable of performing the service, the assembly including components and interfaces specified in an assembly definition and loaded by the assembly manager.

31. The product of claim 30, wherein the service request comprises a name associated with the assembly definition.

32. The product of claim 30, wherein the service request comprises a role name associated with the service.